

A Fuzzy Bi-objective Mathematical Model for Multi-depot Electric Vehicle Location Routing Problem with Time Windows and Simultaneous Delivery and Pick-up

Mozhgan Azarkish¹ & Yousef Aghaeipour²

¹PhD in Industrial Engineering, Tarbiat Modares University, Tehran, Iran. Email: Techra.msn@gmail.com

²PhD in Industrial Engineering, Islamic Azad University, Noor Branch, Mazandaran, Iran. Email: Yousef.a1359@gmail.com



DOI: <http://doi.org/10.38177/AJBSR.2022.4201>

Copyright: © 2022 Mozhgan Azarkish & Yousef Aghaeipour. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Article Received: 07 January 2022

Article Accepted: 19 March 2022

Article Published: 09 April 2022

ABSTRACT

In green logistics, environmentally-friendly vehicles are strongly recommended as a transportation option. One of the green logistics vehicles is the electric vehicle which is a good selection to reduce greenhouse gas emissions. The present paper focused on the location-routing problem in electric vehicles by considering multi-depots and hard and soft time windows in uncertain conditions. We proposed a fuzzy bi-objective mathematical model for electric vehicles with a limitation in charge stations, the dependence of energy consumption to vehicle load, and a simultaneous delivery and pick-up. We used the multi-objectives particle swarm meta-heuristic algorithms based on the Pareto archive and the NSGA-II algorithm to solve this model. To evaluate the validity of the proposed model and algorithms, sample problems of EVRPTW were selected and solved using Gome software and proposed meta-heuristic algorithms. The validation results for the model and algorithm confirmed that the model is valid, and the solving algorithms can solve the model efficiently and converge to an optimal answer. The comparing results of solving algorithms performance showed that, compared to the NSGA-II algorithm, the MOPSO algorithm has a higher ability in all states to generate higher quality responses and more diversity.

Keywords: Routing, Location, Electric vehicles, Simultaneous pick-up and delivery.

1. Introduction

Transportation plays an important role in the economic, production, and service systems, and it is a significant part of the Gross National Product (GNP) for each country. Therefore, researchers have improved routes, eliminated unnecessary trips, and built alternative shortcuts. The Vehicle Routing Problem (VRP) has tried to optimize the products distribution system to optimize related costs and other organization purposes [1]. The progressive urbanization development has made industries, especially support industries, the movement of people and commodities a problem whose complexity is constantly increasing. Each distributor wants to make the most of it, and on the other hand, they face problems such as traffic congestion, air pollution, wasted time on daily travel routes, increased fuel consumption, and vehicle depreciation to distribute commodities and gain liquidity [2]. Transportation has irreversible effects on the environment. For example, we can mention risks such as resource consumption, land use, toxic effects on ecosystems and humans, noise, and greenhouse gas emissions and pollutants. In addition, the emission of greenhouse gases and carbon dioxide is directly related to people's health and indirectly to the ozone layer degradation. Greenhouse gases emitted by transportation cause a significant amount of air pollution in different countries of the world [3]. Among the different transportation methods, 1/3 of greenhouse gases generated by transportation are for diesel vehicles for medium to heavy works [4]. The use of electric vehicles is one of the methods for reducing pollution. They have low emissions and play a considerable role in reducing emissions. Due to the fossil fuels limitation, the use of electric vehicles plays an important role in reducing fossil fuels consumption. Electric vehicles are also more efficient than diesel vehicles. 75% of energy in diesel vehicles is wasted as heat and friction, and only 25% of energy is converted to wheel drive force, while only 20% of energy in electric vehicles is wasted. The number of parts used in electric vehicles is less than that of

conventional vehicles, and the heat generated is also low, so electrical vehicles have less maintenance cost than conventional vehicles.

In some countries, governments provide facilities such as reducing taxes and duties on buying electric vehicles and giving them facilities. As another advantage, electric vehicles can be charged with electricity generated by various resources such as wind energy, solar energy, nuclear energy, water energy, and biofuels. The use of these resources reduces the dependence on oil and gasoline and, as a result, fuel imports are reduced, and it reduces fuel costs. The use of these devices is increasing in the transportation industry (public, commodity, etc.), and many studies have been done on the design and optimization of this type of vehicle. The main limitation in using electric vehicles is a low volume of battery energy compared to fossil-fuel vehicles, long-recharging time, and the limitation in refueling (charging) stations. The refueling in this vehicle is performed by unique equipment, and unlike other gasoline and diesel-fuel vehicles, fuel stations for them are limited. Hence, the limitation of charging stations must be considered for this vehicle.

Another limitation is the dependence of energy consumption on the vehicle load amount that causes limitations in the real world. Due to their positive effect on air pollution reduction, the design and generation of refueling stations are increasing. The reduction of fuel consumption leads to a decrease in service cost and thus customer satisfaction [5]. The present paper focuses on the location-routing problem in electric vehicles. For that, we consider simultaneous multi-depots; the amount of energy consumption depends on the load in electric vehicles, simultaneous pick-up and delivery, and hard and soft time windows. For this purpose, we proposed a two-objective fuzzy-mathematical programming model for electric vehicles with a limitation in charge stations, the dependence of energy consumption to vehicle load, and simultaneous delivery and pick-up. We used the multi-objectives particle swarm meta-heuristic algorithms based on the Pareto archive and the NSGA-II algorithm to solve this model.

2. Literature review

Erdoğan and Miller-Hooks, in 2012, investigated a transportation problem considering the fuel stations. Their proposed model was an integer programming model in which several virtual nodes were defined for each fuel station. That led to an increase in the problem dimensions and the solving solution. In the proposed model, vehicles were considered homogeneous, and the time window equations were used. To solve the problems, they proposed two new large-dimension heuristic algorithms. The first algorithm was based on Clarke and Wright savings algorithm, and the second was based on the distance density clustering algorithm. The computational results of the two algorithms were almost similar [5]. In 2014, Schneider et al. proposed a routing model considering electric vehicles, time windows, and fuel stations. In their model, virtual fuel stations were added. To solve, they applied a combined algorithm consisting of a Local Search (LS) algorithm and a Tabu Search (TS) algorithm. The numerical results obtained in the sample problems were reasonable [6]. In 2014, Philip et al. proposed the electric vehicle routing problem considering the technology of charging at the refueling stations. They also considered the half-full charge in the model. They used Simulated Annealing (SA) and local search algorithms to solve problems. In observations with 200 customers, local search algorithm, and more than 200 customers, the simulated annealing

algorithm had better performance [7]. In 2015, Goeke and Schneider proposed an electrical vehicle routing problem with a time window and considered a mixed fleet that included commercial electric vehicles and conventional vehicles. In this model, the fuel consumption rate was assumed to be a linear function of the traveled distance. Also, fuel consumption equations, speed, cargo slope, and weight were imposed in the limitations. A neighborhood search algorithm was used to solve them [8].

In 2016, Koç and Karaoglan introduced an integer linear programming model where there was no need to add virtual nodes for fuel stations. They modeled the green vehicle routing problem using the definition of auxiliary decision variables. They solved the proposed model using the simulated annealing meta-heuristic algorithm and branch and bound exact algorithm [9]. In 2016, Çatay and Keskin removed the full charge limit, allowing the vehicle to make a full charge. This is more practical in the real world and reduces charging time. They proposed integer programming and used adaptive neighborhood search to solve. Their computational results showed that half-full recharging might improve routing decisions. Furthermore, the computational results showed that their proposed method is more effective in finding solutions compared to other similar research [10]. In 2016, Wen and Roberti considered the routing problem to visit each customer. The time window is included in the proposed model. The proposed integer programming model solved the problems with 46 customers reasonably. They proposed a three-step algorithm based on variable neighborhood search and dynamic programming to obtain quality solutions at the right time. Their computational results show that the proposed algorithm calculates small-size observations in 0.1 sec and generates good solutions for more than 466 customers [11].

In 2016, Hiermann et al. proposed an integer linear programming model for routing problems with mixed transport fleets for electric vehicles. This model assumes that the capacity and cost of using vehicles vary depending on their type. In the model, they considered the time window and charging time limits of each vehicle. They used the branch and price algorithm, a heuristic combination method including large neighborhood search and local search, in addition to the notation process to solve the mode [12].

In 2016, Doppstadt et al. proposed a vehicle model with combined combustion and electric engines. They considered the transportation cost to each node in four states. In the first state, the vehicle is driven by the combustion engine, while the electric engine is charging. The vehicle is driven by combustion and electric engines in the second state. To solve, they used a heuristic 4-step method. Here an initial solution is generated first. Then they used local search and OPT-2 operators in the improvement phase [13]. Schiffer and Walther, in 2017, have studied the electric vehicle routing problem by considering the time window. They simultaneously studied the vehicle routing and the location of charging stations and considered charging different options in attention to the real world limitations. In their model, travel distance, vehicles number, charging stations number, and the total cost was minimized [14]. Camilo Paza et al., in 2018, studied the electric vehicles location-routing problem by considering multi-depots and time windows. They considered the availability of two energy supply technologies, including the conventional charging technology “Plug-In” and the battery replacement stations. They proposed three models. In the first model, conventional charging, in the second model, battery replacement stations, and in the third model, both technologies are considered [15]. Rohmer et al., in 2019, have studied the vehicle

location-routing problem by considering time windows in the delivery system of the final perishable commodities [19]. Almohana et al. (2019) investigated the electronic vehicle routing problem with time windows in uncertain conditions. They presented a single-objective mathematical model and used meta-heuristic algorithms to solve the model [20]. Zheng et al. (2020) studied the vehicle routing problem in multi-warehouse multi-travel states with time windows and distribution time. They aimed to design a set of travels for the fleet of vehicles between different warehouses, to minimize the total travel time [21].

Ghobadi et al., in 2021, studied the electronic vehicles routing problem with fuzzy time windows and multiple depots by considering simultaneous pick-up and delivery. This paper presents an electric vehicle routing problem in a state of multi-warehouses with charging stations by considering an expected penalty for fuzzy time windows on delivery/delivery. Since this problem with fuzzy time windows and delivery/delivery limitations is an np-harp problem, three meta-heuristic algorithms, including simulated annealing, variable neighborhood search, and a combination, were used to solve the model [22]. Much attention has been paid to the electric vehicle location-routing problem in recent years. Whereas in previous research, Camilo Paza et al., in 2018, and Ghobadi et al., in 2021, have considered multi-depots in their research, and in other research, only one depot has been considered the model. On the other hand, attention has not been paid to the dependence of energy consumption on load in electric vehicles and simultaneous pick-up and delivery in previous research. Hence, to cover the existing research gap, the electric vehicle location-routing problem is investigated in the present study by simultaneously considering the multi-depots, the dependent energy consumption amount on the load in an electric vehicle, simultaneous pick-up, and delivery, and hard and soft time window.

Table 1. A summary of previous research

Solving method	Multi-objective	Uncertainty	Simultaneous pick-up and delivery	Time windows	The dependent energy consumption amount on the load	Multiple depots	Routing	Location	Research Topic	Research Researcher/ Date
Heuristic algorithms				*			*		Transportation with considering fuel stations	Erdoğan and Miller-Hooks (2012)
Tabu search				*			*		Electric vehicle routing, with time window and fuel stations	Schneider et al. (2014)
Simulated annealing algorithm				*			*		Electric vehicle routing with considering charge technology in fule stations	Felipe et al. (2014)
Algorithm neighborhood search				*	*		*		Electric vehicle routing with time window	Goeke and Schneider (2015)
Simulated annealing algorithm and branch and bound method							*		Modeling green vehicle routing problem	Koç and Karaoglan (2016)

Solving method	Multi-objective	Uncertainty	Simultaneous pick-up and delivery	Time windows	The dependent energy consumption amount on the load	Multiple depots	Routing	Location	Research Topic	Research Researcher/ Date
Adaptive neighborhood search				*	*		*		Electric vehicle routing with full and half-full charge	Keskin and Çatay (2016)
Based on variable neighborhood search and dynamic planning				*			*		Electric vehicle routing	Roberti and Wen (2016)
Branch and boundary method and a meta-heuristic algorithm based on neighborhood search				*			*		Routing of electric vehicles with a mixed transport fleet	Hiermann et al. (2016)
Heuristic algorithms							*		Electric vehicles routing with the combined combustion and electric engines	Doppstadt et al. (2016)
Heuristic algorithms	*			*			*	*	Electric vehicle location-routing	Schiffer Walther (2017)
Heuristic algorithms				*		*	*	*	Electric vehicle location-routing	Camilo Paza et al. (2018)
Meta-heuristic algorithms		*		*		*	*	*	Electric vehicle location-routing	Almouhanna et al. (2019)
Meta-heuristic algorithms		*	*	*		*	*	*	Location-routing of electric vehicle	Ghobadi et al. (2021)
Gomez software, multi-objective particle swarm meta-heuristic algorithms based on the Pareto archive and the NSGA-II algorithm	*	*	*	*	*	*	*	*	Location-routing of electric vehicles with a time window and multi-depots	Present research

With attention to Table 1, little research has considered multi-depots and the dependent charge consumption amount on the load in electric vehicles location-routing problem. Also, most of the presented models are single-purpose and definitive. While in this study, electric vehicles location-routing problem with multi-depots and

the dependent charge consumption amount on the load in a fuzzy condition has been investigated, and a two-objective mathematical model has been designed for this problem. In addition, in previous research about the electric vehicles location-routing problem, the simultaneous pick-up and delivery of commodities have not been investigated, while this assumption has been taken into account in the present study. Another highlight difference between the present study and previous research is the solving method. In the present study, Gomez software has been used to solve small-size problems, and meta-heuristic multi-objective particle swarm algorithms based on Pareto archive and NSGA-II have been used to solve the large-size problems. The variable neighborhood search method improves the solutions in the proposed algorithm structure.

3. Mathematical modeling

This study investigated the electrical vehicles location-routing problem by considering multi-depots and time windows. This problem consists of multi-vehicles, several charging stations, and multi-depots which charging stations and depots are located. Here, the charge is considered fully, and the fuel consumption rate depends on the load that vehicles carry in each route. It means that the fuel consumption amount in a vehicle will be different at a given consumption rate on an unload route and the same route with a load. Hence, unlike previous research, we have considered the fuel consumption of electric vehicles to depend on the load carried. Considering this assumption can play an important role in choosing the optimal route and certainly has an effective role in visiting or not visiting fuel stations. In this model, simultaneous pick-up and delivery are assumed, and each point has two types of pick-up and delivery requests that are considered fuzzy. Also, service to demand points must be performed in a predetermined time window.

The assumptions of the studied model are as follows:

- Vehicles have a limited capacity.
- The charge consumption by each vehicle depends on the vehicle load.
- Some parameters are considered definite in this model, and some are indefinite.
- The combined routing problem includes customers, multi-warehouses (depots), and a set of stations to charge the battery.
- The location of depots and charging stations is considered.
- Each charging station is allowed not to be visited at all or more than once.
- Vehicles are needed to meet all customers. All customers must be met.
- There are two types of time windows, hard and soft. Services that are outside the hard period and within the soft period, $[Lb_i, Ub_i]$, (in terms of delays) are subject to a specified and fixed penalty cost per unit time.
- Service outside the soft period is not allowed.
- Pick-up and delivery are simultaneously considered.
- The model is as two-objective.

3.1. Model indices

A set of all nodes, including customers, depots, and charging stations

N_c : A set of customers, c is customer indice.

N_f : A set of potential nodes of charging stations, f is indice of charging station.

N_d : A set of potential nodes of depots, d is the indice of the depot.

K : Number of vehicles, k is indice of vehicle.

3.2. Model parameters

d_{ij} : The distance between nodes i and j .

c_k : The variable cost of using a k^{th} vehicle per unit of distance.

f_c : Fixed cost of using a vehicle.

c_0 : Cost per unit of charge.

ρ_0 : Charge consumption rate per unit of distance.

$\tilde{\alpha}$: Charge consumption rate depending on the vehicle load per unit of distance.

W_p : Weight per unit of each delivered and received product.

\tilde{d}_c : The amount of fuzzy demand delivered to the c^{th} customer.

\tilde{p}_c : The amount of fuzzy demand received from the c^{th} customer.

cap_k : Load fuzzy capacity of the k^{th} vehicle (in kilograms).

\tilde{Q} : The battery fuzzy capacity for any vehicle.

t_{ij} : Travel time between nodes i and j .

M : A selected large number.

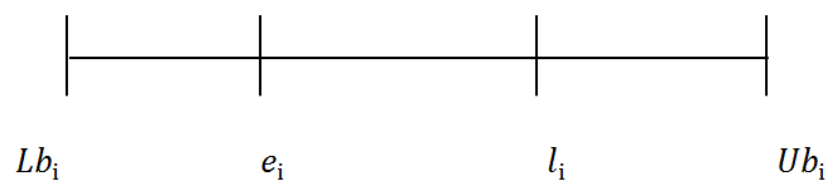
\tilde{P}_i : Fuzzy penalty amount for any delay.

$[e_i, l_i]$: Hard time window for i^{th} node.

$[Lb_i, Ub_i]$: Soft time window for i^{th} node.

f_f : The establishing cost for a charging station at point f .

f_d : The establishing cost for a depot at node d .



3.3. Model variables

z_f : If the charging station is located at node f , it will be equal to 1. Otherwise, it will be equal to 0.

z_d : If the depot is located at node d , it will be equal to 1; otherwise, it will be 0.

z_k : If k^{th} vehicle is used, it will be equal to 1, and otherwise, it will be equal to 0.

x_{ij}^{kd} : If k^{th} vehicle, which started its travel from depot d , travels between two nodes i and j , it is equal to 1; otherwise, it is equal to 0.

y_{ij}^{kd} : The amount of demand delivered to the customer in node j , if the k^{th} vehicle travels between nodes i and j (the vehicle that started its travel from depot d).

z_{ij}^{kd} : The amount of demand received from the customer in node j if the k^{th} vehicle travels between nodes i and j .

a_i : The time which a vehicle reaches the location of each node i .

p_i : The time which a vehicle leaves the location of each node i .

St_i : The service time for each node i .

y_{ck} : The charge amount that the k^{th} vehicle has when meeting the c^{th} customer.

y_{fk} : The charge amount that the k^{th} vehicle has when it reaches the f^{th} charging station.

w_{ij}^k : The load amount that k^{th} vehicle carries between nodes i and j .

Yl_i : The delay in servicing to node i

3.4. The main structure of the mathematical model

$$\min z1 = \sum_{f=1}^{N_f} f_f \times z_f + \sum_{d=1}^{N_d} f_d \times z_d + \sum_{k=1}^K f_c \times z_k + \sum_{k=1}^K \sum_{d=1}^{N_d} \sum_{i=1}^N \sum_{j=1}^N (C_k d_{ij} x_{ij}^{kd} + c_0 \times d_{ij} (\widetilde{\rho}_0 + \widetilde{\alpha} w_{ij}^k) x_{ij}^{kd}) \quad (1)$$

The objective Equation (1): The minimizing costs

$$\min z2 = \sum_{i=1}^n \widetilde{P}l_i Yl_i \quad (2)$$

The objective Equation (2): The minimizing delays in providing or receiving service to each node.

$$\sum_{d=1}^{N_d} \sum_{i=1}^N \sum_{k=1}^K x_{ij}^{kd} = 1 \quad \forall j \in N_c \quad (3)$$

Limitation (3) ensures that customer j is certainly met.

$$\sum_{j=1}^N \sum_{d=1}^{N_d} \sum_{k=1}^K x_{ij}^{kd} = \sum_{j=1}^N \sum_{d=1}^{N_d} \sum_{k=1}^K x_{ji}^{kd} \quad \forall i \in N \quad (4)$$

Limitation (4) shows that vehicles entering a node will leave that node, certainly.

$$\sum_{j=1}^N x_{dj}^{kd} = \sum_{j=1}^N x_{jd}^{kd} \quad \forall d \in N_d, \forall k \in K \quad (5)$$

Limitation (5) shows that vehicles that started their travel from depot d will certainly return to the same depot at the travel end.

$$\sum_{j=1}^N x_{ij}^{kd} \leq 1 \quad \forall i \in N_f, d \in N_d, \quad \forall k \in K \quad (6)$$

Limitation (6) shows that a charging station may be met or not.

$$\sum_{j=1}^N \sum_{d=1}^{N_d} \sum_{k=1}^K x_{ji}^{kd} y_{ji}^{kd} - \sum_{j=1}^N \sum_{d=1}^{N_d} \sum_{k=1}^K x_{ij}^{kd} y_{ij}^{kd} = \tilde{d}_i \quad \forall i \in N_c \quad (7)$$

$$\sum_{j=1}^N \sum_{d=1}^{N_d} \sum_{k=1}^K x_{ij}^{kd} z_{ij}^{kd} - \sum_{j=1}^n \sum_{d=1}^{N_d} \sum_{k=1}^K x_{ji}^{kd} z_{ji}^{kd} = \tilde{p}_i \quad \forall i \in N_c \quad (8)$$

The two limitations (7) and (8) ensure that the delivery/pick-up request for every i^{th} customer are met.

$$\sum_j x_{ji}^{kd} w_{ji}^k + \sum_{j=1}^n \sum_{k=1}^K w_p x_{ji}^{kd} y_{ji}^{kd} = \sum_{j=1}^n \sum_{k=1}^K w_p x_{ji}^{kd} z_{ji}^{kd} + \sum_{j=1}^N x_{ij}^{kd} w_{ij}^k \quad \forall i \in N, d \in N_d \quad (9)$$

Limitation (9) calculates the vehicle load amount.

$$w_{ij}^k \leq \tilde{c} \tilde{a} p_k \times x_{ij}^{kd} \quad \forall i, j \in N, d \in N_d, k \in K \quad (10)$$

Limitation (10) ensures that the vehicle load does not exceed its capacity.

$$d_{ij}(\tilde{\rho}_0 + \tilde{a} w_{ij}^k) x_{ij}^{kd} - \tilde{Q}(1 - x_{ij}^{kd}) \leq y_{ik} - y_{jk} \quad \forall i \in N_c, j \in N_c, d \in N_d, k \in K \quad (11)$$

$$y_{ik} - y_{jk} \leq d_{ij}(\tilde{\rho}_0 + \tilde{a} w_{ij}^k) x_{ij}^{kd} + \tilde{Q}(1 - x_{ij}^{kd}) \quad \forall i \in N_c, j \in N_c, d \in N_d, k \in K \quad (12)$$

$$d_{ij}(\tilde{\rho}_0 + \tilde{a} w_{ij}^k) x_{ij}^{kd} - \tilde{Q}(1 - x_{ij}^{kd}) \leq y_{ik} - y_{jk} \quad \forall i \in N_c, j \in N_f, d \in N_d, k \in K \quad (13)$$

$$y_{ik} - y_{jk} \leq d_{ij}(\tilde{\rho}_0 + \tilde{a} w_{ij}^k) x_{ij}^{kd} + \tilde{Q}(1 - x_{ij}^{kd}) \quad \forall i \in N_c, j \in N_f, d \in N_d, k \in K \quad (14)$$

Limitations (11) to (14) calculate the residual charge amount at each node.

$$y_{ik} \geq d_{id}(\tilde{\rho}_0 + \tilde{a} w_{id}^k) x_{id}^{kd} \quad \forall i \in N_c, d \in N_d, k \in K \quad (15)$$

Limitation (15) ensures that the vehicle is sufficiently charged to return to the depot.

$$y_{ik} = \tilde{Q} \quad \forall i \in N_d \cup N_f, \quad \forall k \in K \quad (16)$$

Limitation (16) ensures that the vehicle receives charge according to its capacity when meeting the charging station, and each vehicle leaves the depots on a full charge.

$$a_j \geq (a_i + St_i) - \left((1 - x_{ij}^{kd})M\right) + t_{ij}, \forall k \in K, \quad i, j \in N, d \in N_d \quad (17)$$

$$a_j \leq (a_i + St_i) + \left((1 - x_{ij}^{kd})M\right) + t_{ij}, \quad \forall k \in K, \quad i, j \in N, d \in N_d \quad (18)$$

Limitations (17) and (18) are related to the time window and feasibility of the schedule of each node, and if node j is not serviced after node i , these two limitations are disabled.

$$P_1 = 0 \quad (19)$$

$$Lb_i \leq a_i \leq Ub_i \quad \forall i \in N \quad (20)$$

$$p_i \geq St_i + a_i \quad \forall i \in N \quad (21)$$

$$a_i - l_i \leq Yl_i \quad \forall i \in N \quad (22)$$

Number (19) shows that the origin leaving time is at the moment 0. Limitations (20) and (21) show that the arrival, leaving, and service times for each node must be within a soft time window. Limitation (22) express the delay amount in service for any node.

$$x_{ij}^{kd} \leq Mz_d \quad \forall k \in K, \quad i, j \in N, d \in N_d \quad (23)$$

Limitation (23) ensures a vehicle starts its travel from a depot to cross the edges if that depot is established.

$$x_{ij}^{kd} \leq Mz_f \quad \forall k \in K, \quad i \text{ or } j \in N_f, d \in N_d \quad (24)$$

$$y_{ik} \leq Mz_f \quad \forall k \in K, \quad i \in N_f, k \in K \quad (25)$$

Limitations (24) and (25) ensure that a vehicle meets a charging station if that station is established.

$$x_{ij}^{kd}, z_d, z_f = \{0,1\}, y_{ik}, z_{ij}^{kd}, y_{ij}^{kd}, w_{ij}^k, a_i \geq 0 \quad \forall i, j \in N, d \in N_d \quad (26)$$

Limitation (26) shows the values range of the model decision variables.

According to expressions (1) and (2), the first objective function includes minimizing the costs of locating depots, charging stations, and using vehicles. Minimizing these costs means making up fewer depots and charging stations and using fewer vehicles. In this state, the service level or in-time services decreases and causes delay. The delay increases with a decrease in the cost of locating depots, charging stations, and using vehicles. Decreasing the first objective function increases the second objective function, which these two objective functions conflict each other.

3.5. Defuzzification, the model

Many methods have been proposed to solve fuzzy mathematical programming problems. The method of ranking proposed by Jimenez et al. in 2007 is used in this research. Jimenez proposed a method of ranking fuzzy numbers by comparing their waiting intervals [16]. If a triangular fuzzy number is as $\tilde{A} = \{L, M, U\}$, it can be written as follows (Fig.1):

$$\mu_A(x) = \begin{cases} f_A(x) = \frac{x-L}{M-L} & L \leq x \leq M \\ 1 & x = M \\ g_A(x) = \frac{x-L}{U-M} & M \leq x \leq U \end{cases} \quad (27)$$

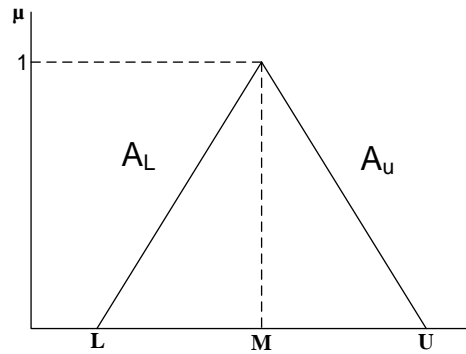


Fig.1. Triangular fuzzy number

To ensure that the inverse of functions $f_A(x)$ $g_A(x)$ exists, it is assumed $f_A(x)$ to be continuous and ascending and $g_A(x)$ continuous and descending. The expected range of a fuzzy number is defined as follows:

$$EI(\tilde{A}) = [E_1^{\tilde{A}}, E_2^{\tilde{A}}] = \left[\int_{a_1}^{a_2} x df_A(x) - \int_{a_3}^{a_4} x dg_A(x) \right] \quad (28)$$

By aggregating the components and changing the variable, we will have:

$$EI(\tilde{A}) = [E_1^{\tilde{A}}, E_2^{\tilde{A}}] = \left[\int_0^1 f_A^{-1}(\alpha) d\alpha - \int_0^1 g_A^{-1}(\alpha) d\alpha \right] \quad (29)$$

If the functions $f_A(x)$ $g_A(x)$ are linear and \tilde{A} a triangular fuzzy number, the expected range will be as follows:

$$EI(\tilde{A}) = \left[\frac{1}{2}(L + M), \frac{1}{2}(M + U) \right] \quad (30)$$

Also, the expected value of the fuzzy number \tilde{A} is half the value of the expected range.

$$EV(A) = \frac{E_1^{\tilde{A}} + E_2^{\tilde{A}}}{2} \quad (31)$$

Furthermore, for a triangular fuzzy number, \tilde{A} it is as follows:

$$EV(A) = \frac{L + 2M + U}{2} \quad (32)$$

Definition 1. For both fuzzy numbers \tilde{A} and \tilde{B} , and membership degree for \tilde{A} greater than \tilde{B} is as follow:

$$\mu_M(\tilde{A}, \tilde{B}) = \begin{cases} 0 & \text{if } E_2^A - E_1^B < 0 \\ \frac{E_2^A - E_1^B}{E_2^A - E_2^B - (E_1^A - E_1^B)} & \text{if } 0 \in [E_1^A - E_2^B, E_2^A - E_1^B] \\ 1 & \text{if } E_1^A - E_2^B > 0 \end{cases} \quad (33)$$

So that $[E_1^A, E_2^A]$ and $[E_1^B, E_2^B]$ are the expected ranges for \tilde{A} and \tilde{B} . When $\mu_M(\tilde{A}, \tilde{B}) = 0.5$, \tilde{A} and \tilde{B} are the same.

When $\mu_M(\tilde{A}, \tilde{B}) \geq \alpha$, \tilde{A} with the least, a degree α is greater than or equal to \tilde{B} , it can be written as $\tilde{A} \geq_\alpha \tilde{B}$

Definition 2. Suppose $x \in R^n$ is a vector; it is acceptable with degree α if $\min\{\mu_M(\tilde{A}x, \tilde{B})\} = \alpha$ it can be written as $\tilde{A}x \geq_\alpha \tilde{B}$. Equation (31) can be rewritten as follows:

$$\frac{E_2^{Ax} - E_1^B}{E_2^{Ax} - E_1^{Ax} + E_2^B - E_1^B} \geq \alpha \quad (34)$$

or

$$[(1 - \alpha)E_2^A + \alpha E_1^A]x \geq \alpha E_2^B + (1 - \alpha)E_1^B$$

Thus, according to the above definitions, we can convert the fuzzy model into a definite and accurate model. As follows:

$$\begin{aligned} & \text{MinEV}(\tilde{C})x \\ & \text{s.t. :} \\ & x \in \{x \in R^n \mid \tilde{A}x \geq_\alpha \tilde{B}, x \geq 0\} \end{aligned} \quad (35)$$

Now, according to the above definitions and using the mentioned method, we convert the proposed fuzzy programming model into a definite model like it:

Definite model of objective functions:

$$\begin{aligned} \min z1 = & \sum_{f=1}^{N_f} f_f \times z_f + \sum_{d=1}^{N_d} f_d \times z_d + \sum_{k=1}^K f_c \times z_k \\ & + \sum_{k=1}^K \sum_{d=1}^{N_d} \sum_{i=1}^N \sum_{j=1}^N (C_k d_{ij} x_{ij}^{kd} + c_0 \times d_{ij} (\frac{\rho_0^L + 2\rho_0^M + \rho_0^U}{2} \\ & + \frac{\alpha^L + 2\alpha^M + \alpha^U}{2} w_{ij}^k) x_{ij}^{kd}) \end{aligned} \quad (36)$$

$$\min z2 = \sum_{i=1}^n \frac{PL_i^L + 2PL_i^M + PL_i^U}{2} Yl_i \quad (37)$$

Definite model of limitations (7) and (8) with β -section:

$$\sum_{j=1}^N \sum_{d=1}^{N_d} \sum_{k=1}^K x_{ji}^{kd} y_{ji}^{kd} - \sum_{j=1}^N \sum_{d=1}^{N_d} \sum_{k=1}^K x_{ij}^{kd} y_{ij}^{kd} = (1 - \beta) \frac{d_i^L + d_i^M}{2} + \beta \frac{d_i^M + d_i^U}{2} \quad \forall i \in N_c \quad (38)$$

$$\sum_{j=1}^N \sum_{d=1}^{N_d} \sum_{k=1}^K x_{ij}^{kd} z_{ij}^{kd} - \sum_{j=1}^n \sum_{d=1}^{N_d} \sum_{k=1}^K x_{ji}^{kd} z_{ji}^{kd} = (1 - \beta) \frac{p_i^L + p_i^M}{2} + \beta \frac{p_i^M + p_i^U}{2} \quad \forall i \in N_c \quad (39)$$

Definite model of limitation (10) with β -section:

$$w_{ij}^k \leq [(1 - \beta) \frac{cap_k^L + cap_k^M}{2} + \beta \frac{cap_k^M + cap_k^U}{2}] \times x_{ij}^{kd} \quad \forall i, j \in N, d \in N_d, k \in K \quad (40)$$

Definite model of limitations (11-14) with β -section:

$$d_{ij}((1 - \beta) \frac{\rho_0^L + \rho_0^M}{2} + \beta \frac{\rho_0^M + \rho_0^U}{2} + [(1 - \beta) \frac{\alpha^L + \alpha^M}{2} + \beta \frac{\alpha^M + \alpha^U}{2}] w_{ij}^k) x_{ij}^{kd} - [(1 - \beta) \frac{Q^L + Q^M}{2} + \beta \frac{Q^M + Q^U}{2}] (1 - x_{ij}^{kd}) \leq y_{ik} - y_{jk} \quad \forall i \in N_c, j \in N_c, d \in N_d, k \in K \quad (41)$$

$$y_{ik} - y_{jk} \leq d_{ij}((1 - \beta) \frac{\rho_0^L + \rho_0^M}{2} + \beta \frac{\rho_0^M + \rho_0^U}{2} + [(1 - \beta) \frac{\alpha^L + \alpha^M}{2} + \beta \frac{\alpha^M + \alpha^U}{2}] w_{ij}^k) x_{ij}^{kd} + [(1 - \beta) \frac{Q^L + Q^M}{2} + \beta \frac{Q^M + Q^U}{2}] (1 - x_{ij}^{kd}) \quad \forall i \in N_c, j \in N_c, d \in N_d, k \in K \quad (42)$$

$$d_{ij}((1 - \beta) \frac{\rho_0^L + \rho_0^M}{2} + \beta \frac{\rho_0^M + \rho_0^U}{2} + [(1 - \beta) \frac{\alpha^L + \alpha^M}{2} + \beta \frac{\alpha^M + \alpha^U}{2}] w_{ij}^k) x_{ij}^{kd} - [(1 - \beta) \frac{Q^L + Q^M}{2} + \beta \frac{Q^M + Q^U}{2}] (1 - x_{ij}^{kd}) \leq y_{ik} - y_{jk} \quad \forall i \in N_c, j \in N_f, d \in N_d, k \in K \quad (43)$$

$$y_{ik} - y_{jk} \leq d_{ij}((1 - \beta) \frac{\rho_0^L + \rho_0^M}{2} + \beta \frac{\rho_0^M + \rho_0^U}{2} + [(1 - \beta) \frac{\alpha^L + \alpha^M}{2} + \beta \frac{\alpha^M + \alpha^U}{2}] w_{ij}^k) x_{ij}^{kd} + [(1 - \beta) \frac{Q^L + Q^M}{2} + \beta \frac{Q^M + Q^U}{2}] (1 - x_{ij}^{kd}) \quad \forall i \in N_c, j \in N_f, d \in N_d, k \in K \quad (44)$$

Definite model of limitation (15) with β -section:

$$y_{ik} \geq d_{ij}((1 - \beta) \frac{\rho_0^L + \rho_0^M}{2} + \beta \frac{\rho_0^M + \rho_0^U}{2} + [(1 - \beta) \frac{\alpha^L + \alpha^M}{2} + \beta \frac{\alpha^M + \alpha^U}{2}] w_{ij}^k) x_{ij}^{kd} \quad \forall i \in N_c, d \in N_d, k \in K \quad (45)$$

Definite model of limitation (16) with β -section:

$$y_{ik} = (1 - \beta) \frac{Q^L + Q^M}{2} + \beta \frac{Q^M + Q^U}{2} \quad \forall i \in N_d \cup N_f, \forall k \in K \quad (46)$$

4. The solving method

The present study is focused on modeling and solving the “multi-depots location-routing problem” in electric vehicles by considering a simultaneous pick-up and delivery and time window and a multi-objective particle swarm algorithm based on the Pareto archive was proposed to solve the model. To continue, the proposed structure of the combined multi-objective PSO method was considered to optimize the two objective functions in the model simultaneously. The aim of designing the above method was to achieve more global optimal solutions (or more Pareto solutions). To evaluate this algorithm, its output was compared with the NSGA-II algorithm. PSO was first developed by a social psychologist named James Kennedy and an electronic engineer named Russell C. Eberhart in

1995. This development was based on previous experiences in modeling group behaviors observed in many birds' species. The general structure designed for the combined multi-objective PSO method is in the following

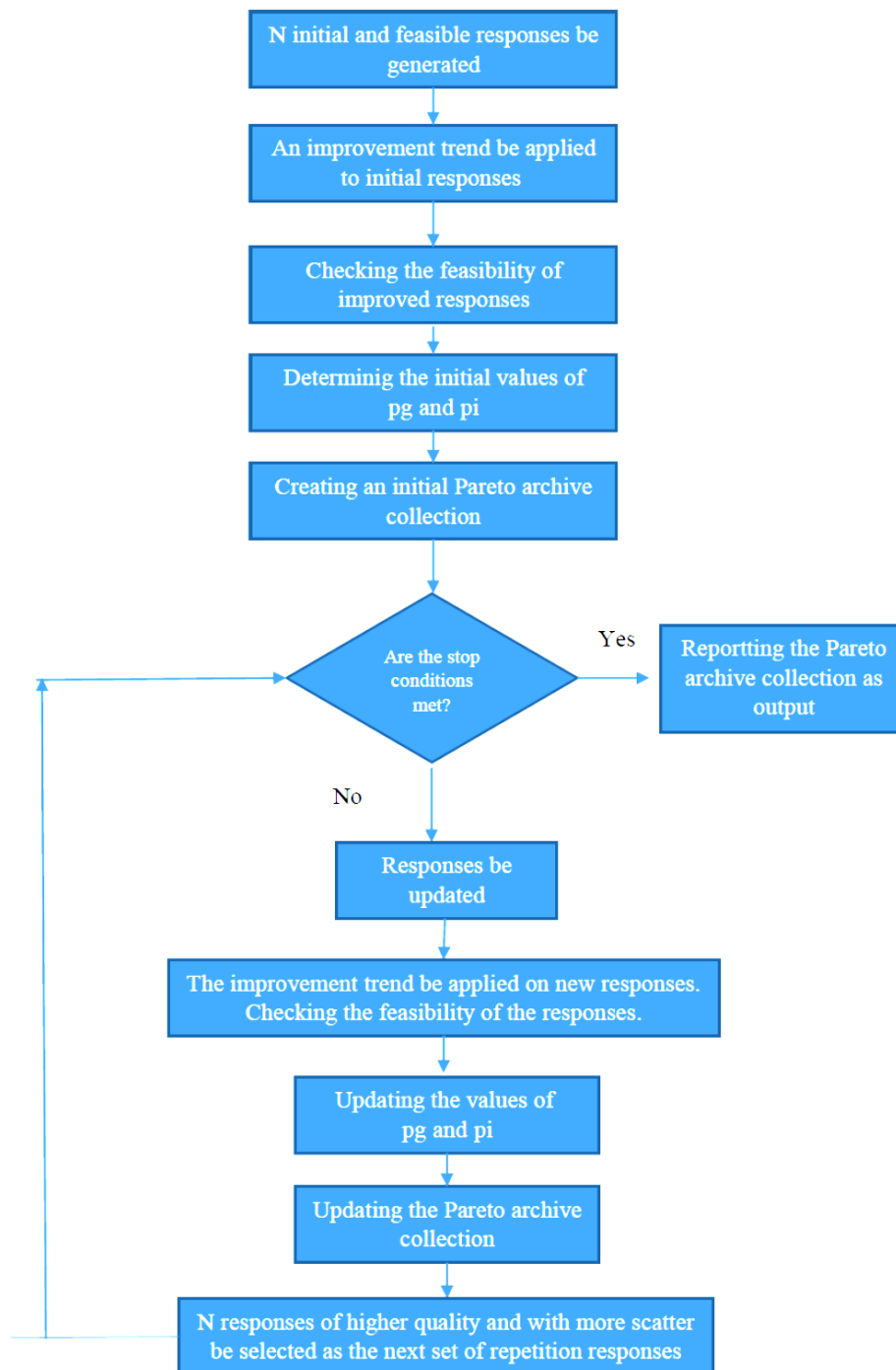


Fig.2. Diagram of steps of multi-objectives particle swarm algorithm based on the Pareto archive

4.1. How to display the response

Many matrices were used to display each solution in this research, and each matrix corresponded to a variable defined in the model. For example, a one-dimensional matrix with N_f (number of charging stations) was used to display the variable z_f , a one-dimensional matrix with N_d (number of depots) was used to display the variable z_d , and a one-dimensional matrix with N_k (number of vehicles) was used to display the variable z_k .

4.2. Generation of the initial solutions

A random approach is used to generate the initial solutions in this paper. To generate the initial solutions, the Z_f , Z_d , and Z_k matrices were randomly generated, and then the rest of the solution matrices (model variables) were quantified in attention to the model constraints. If the population size is N , whenever a solution was generated as described, if not repeated, it will be added to the solution population. This trend will continue until the solutions number in the population reaches $\alpha \times N$ (α is a number greater than 1). The solutions generating method stops after $\alpha \times N$ repetition.

On the other hand, the solutions number in each algorithm repetition equals N . Hence; we have to choose N solutions from $\alpha \times N$ available solutions as initial production orders. In this paper, the selection of the solutions initial population was based on a rapid method of sorting non-dominated solutions, described by Deb et al. in 2002 [17]. In this method, $\alpha \times N$ solutions designed by the mentioned algorithm are sorted and leveled. Each level number indicates the quality of the solutions at that level. For example, the quality of the solution in level 1 is higher than the quality of the solution in level 2. Then, for the solutions in each level, a scale called the crowding distance was calculated in proportion to that level. This scale for the solutions of each level indicates the scatter of the solutions of that level. In this paper, a criterion called C_s is defined for selecting the initial responses, which is obtained by the following Equation [18]

$$C_s = \frac{\text{rank}}{\text{crowding_dis}} \quad (47)$$

The above criterion is calculated for each of the available responses.

Rank: Indicates the level number where the response is located.

Crowding dis: The crowding distance of each response is proportional to the rank of that response.

After calculating the above criterion for all the solutions, the solutions were arranged ascending (based on the value of C_s), and the first N solutions with fewer values of C_s were selected as the initial algorithm solutions. The use of the C_s criterion is based on selecting solutions with higher quality and more scatter as the initial population. The solutions generated are improved as much as possible using the improvement procedure. How the improvement procedure works will be described in the next section.

4.3. Improvement procedure

In this paper, an improvement procedure is designed that applies to the solutions (particles) and improves those solutions (particles) as much as possible. The designed improvement procedure in this research varies based on variable neighborhood search. In this procedure, three neighborhood search operators are designed combined with the VNS structure. This combination is as follows [18]:

{for each input solution s

$K=1$

While the stopping criterion is met, **do**

S1=Apply NSS type k

S=Acceptance method (S,S1)

If s is improved **then**

K=1

Else

K=k+1

If k=4 **then**

K=1

Endif

End while}

As shown in the above structure, after applying the neighborhood structure to the solution, the acceptance procedure is applied to the resulting solution and the previous solution, and one of the two solutions is selected as the next repetition solution for VNS. The acceptance procedure determines and selects the dominant solution from two solutions using non-dominated relations. First neighborhood search structure: In this structure, the indices f_1 and f_2 are randomly generated in a uniform range $[1, N_f]$, and the values of the cells corresponding to the indices generated in the matrix related to the variable Z_f are exchanged with each other. Second neighborhood search structure: In this structure, the indices d_1 and d_2 are randomly generated in a uniform range $[1, N_d]$, and the values of the cells corresponding to the indices generated in the matrix related to the variable Z_d are exchanged with each other. Third neighborhood search structure: In this structure, the indices k_1 and k_2 are randomly generated in a uniform range $[1, K]$, and the values of the cells corresponding to the indices generated in the matrix related to the variable Z_k are exchanged with each other.

4.4. Particle update

Here, genetic algorithm operators were used to update the particles [18]. How to update the particles is as follows:

$$x_i^{t+1} = (x_i^t - p_i^t) + (x_i^t - p_g^t) + \overline{x_i^t} \quad (48)$$

In the above Equation:

x_i^{t+1} : i^{th} particle in $(t + 1)^{\text{th}}$ repetition (generation)

x_i^t : i^{th} particle in t^{th} repeating.

p_i^t : The best response which i^{th} particle reaches it.

p_g^t : The best-found response.

$\overline{x_i^t}$: A neighbor generated by a mutation operator for x_i^t .

‘-’ : This symbol shows the crossover operator.

’+‘ : This is a selection symbol.

To obtain the i^{th} response in the $(t+1)^{\text{th}}$ repetition, five responses are generated: two responses are obtained from crossover operator between x_i^t and P_i^t , two are obtained from crossover operator on x_i^t and P_g^t and one obtained from mutation operator on x_i^t . Finally, the response with higher quality and more scatter among these five responses is selected as x_i^{t+1} . In fact, P_g^t and P_i^t are used in this Eq. as a guide to reach next repetition responses.

Crossover operator:

The crossover operator designed in this algorithm is a single point crossover operator that is applied to the variables Z_f , Z_d , and Z_k . After two parents are input to the crossover operator, an index for each variable is randomly generated in a uniform range, and then two children are formed from the crossover of the two parents. Mutation Operator: The mutation operator used in the above Equation for particle updating is the same as variable neighborhood search, which is fully described in the previous section.

4.5. Updating p_i^t and p_g^t

For each i^{th} particle, among the neighborhoods found for this response, if there is a better neighborhood than p_i , p_i will be replaced by it. Otherwise, it remains unchanged. Of all the responses found so far, if the best response is better than p_g , it will replace p_g . Otherwise, it remains unchanged.

4.6. Updating the Pareto archive

The solution method used in the present research is based on the Pareto archive. A set called the Pareto archive is considered in the proposed algorithm, which holds the non-dominated responses produced by the algorithm. This set will be updated each time the algorithm is repeated. The method of updating is that the generated responses in it are repeated, and the responses existing in the Pareto archive are poured into a response pool and leveled together. Then among these responses, the responses existing in the first level (non-dominated responses) are selected and considered as the new Pareto archive.

4.7. Select next-generation responses

In each repeating, the algorithm needs a population of responses. To select the next repetition population in this research, the existing responses in its population are repeated, and the new responses generated by the algorithm are poured together in a response pool. After leveling and calculating the crowding distance criterion for each response according to the level of that response, using the Deb rule in 2002, N responses with the highest quality and the most scatter are selected as the repetition population after the algorithm.

4.8. NSGA-II algorithm

In this research, in order to solve the model, in addition to the multi-objective particle swarm algorithm, the NSGA-II algorithm based on the Pareto archive is also used. The diagram of the proposed structure of the NSGA-II algorithm is as follows.

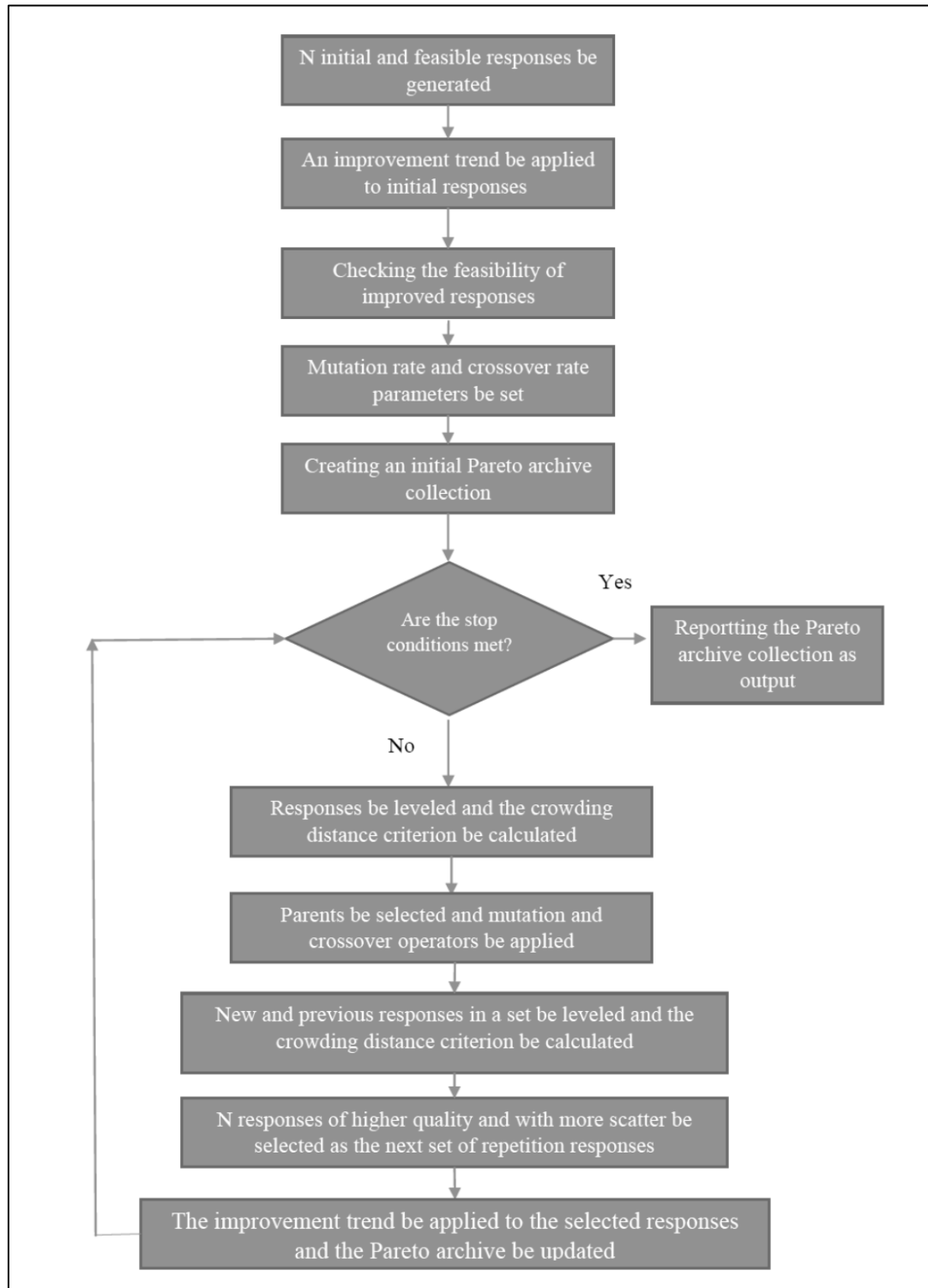


Fig.3. Diagram of NSGA-II algorithm steps

How to generate the initial responses in this algorithm is random. N feasible responses will be randomly generated as the first repetition population of the algorithm. To apply the crossover operator, first, the parents must be selected for the crossover. In the present study, two higher quality responses are selected to select the parents using the non-dominated Equation and the double tournament method. The NSGA-II algorithm has two operators, crossover and mutation. The mutation and crossover operators used in this research are the same mutations and crossover operators used in particle updates in the particle swarm algorithm. This similarity is that the comparison

of the two algorithms is impartial. The particle swarm algorithm and NSGA-II algorithm can be compared under the same conditions.

4.9. Dominant relation

Due to the conflict between objective functions in the present research, using a multi-criteria optimization approach to solve the problem is more appropriate than single-criteria optimization methods. Hence, descriptions of the principles and foundations of multi-criteria optimization are provided in the following. The main difference between multi-criteria and single-criterion optimization problems is the existence of different and conflicting objective functions in multi-criteria problems. The existence of such goals makes it impossible to achieve the optimal solution (or solutions) through conventional algorithms for single-criteria optimization problems. To better understand the nature of multi-criteria optimization problems, the following principal definitions are provided (in all definitions, the minimization model below is considered with p decision variable and q objective function).

$$x, y \in R^p, q \geq 1 \quad \text{Min } y=f(x)=(f_1(x), f_2(x), \dots, f_q(x))$$

Definition 1: Domination Relation

In multi-criteria optimization problems, the vector x_1^p overcomes the vector x_2^p , if the following two conditions are met:

$$\begin{aligned} f_i(x_1^p) &\leq f_i(x_2^p) & i=1, \dots, q \\ f_i(x_1^p) &< f_i(x_2^p) & \exists \quad i=1, \dots, q \end{aligned}$$

Therefore, the main goal in such problems is to find a set of points that dominate other points.

4.10. Sorting and leveling responses

The NSGA-II algorithm sets and levels the response in the population based on non-dominated relations. Relationships defined in section 9.4 were used to set and level the responses. First, all responses were compared using the mentioned relations, and the responses that no response can overcome were considered first-level responses. Then, this trend was repeated for the set of unspecified responses, and the next levels were specified. A response is of better quality when its level number is low.

Therefore, levels with lower numbers were first used to select the responses. If it is possible to choose between two responses on the same level, the crowding distance criterion is used. The priority of a response for selection is high when the value of the crowding distance criterion for that response in a level is high.

5. Computational Results

First, to check the validity of the model and algorithm, the model of a small-size sample problem was solved by the Gomez software and multi-objective particle swarm algorithm, and then their results were compared. After validating the model and algorithm to examine the performance of MOPSO and NSGA-II algorithms, we performed them in the MATLAB software. The results of its performance in experimental problems generated by

them were compared based on comparative indices of quality, uniformity, scatter, and solving time. All calculations were performed using the core i7 7500U-12GB-1TB -R5 M335 4GB computer.

5.1. Comparative indices

There are many different indices for evaluating the quality and scatter of multi-objective meta-heuristic algorithms. For comparison in this paper, three indices, including quality, uniformity, and dispersion, are considered [18]. They are described in the following.

Quality index: This index compares the quality of Pareto responses obtained by each method. This index levels all Pareto responses obtained by artificial bee colony algorithm and genetic algorithm together and determines what percentage of level 1 belongs to each method. An algorithm has higher quality when this percentage is higher for it.

Uniformity index: This index checks the distribution uniformity of Pareto responses obtained at the boundaries of the responses. This index is defined as follows:

$$s = \frac{\sum_{i=1}^{N-1} |d_{mean} - d_i|}{(N-1) \times d_{mean}} \quad (49)$$

In the above Equation, d_i represents the Euclidean distance between two adjacent non-dominated responses, and d_{mean} is the average of d_i values.

Scatter index: This index is used to determine the amount of non-dominated responses found on the optimal boundary. The definition of scattering index is as follows

$$D = \sqrt{\sum_{i=1}^N \max(\|x_t^i - y_t^i\|)} \quad (50)$$

In the above Equation, $\|x_t^i - y_t^i\|$ represents the Euclidean distance between two adjacent responses x_t^i and y_t^i on the optimal boundary.

5.2. Experimental problems

In this paper, several experimental problems are selected from the benchmark data of the electric vehicle routing problem. Since the benchmark sample problems do not fit the model presented in this research, some parameters have been adjusted based on existing data and previous research to check the model validity. In this paper, three small-size problems based on the C101 problem were designed from EVRPTW sample problems and solved using the Gomez software and the proposed algorithm. To design the problems from problem C101, several stations are considered depots and a number as stations, which number of depots, number of stations, and number of customers is different for each problem with small size.

Table 2. Small-size problems

Number of customers	Number of stations	Number of depots	No
5	3	3	1
10	3	3	2
15	3	3	3

After evaluating the model validity and the solution algorithm, other problems in the EVRPTW benchmark with 100 customers were selected and solved by a multi-objective particle swarm algorithm and NSGA-II algorithm. Then, the performance of the two algorithms was compared based on the described indices.

5.3. Setting the parameters

The algorithm parameters are set as follows:

In a multi-objective particle swarm algorithm, the population size is considered 500, the number of repetitions in the VNS algorithm, 10, and the number of repetitions of the algorithm, 500. In the genetic algorithm, the rates of 0.8 and 0.1 are considered for the crossover and mutation, respectively, and the population size is equal to 500, and the algorithm repetition number is 500. To set the model parameters, the parameters in the EVRPTW benchmark are used, and the other parameters in the model are set as follows: To generate triangular numbers related to each fuzzy parameter (m_1, m_2, m_3), first, m_2 is selected, then a random number (r) is generated in the range (0,1). The m_1 and m_3 will be generated using the Equation $m_2*(1-r)$ and $m_2*(r+1)$, respectively. To quantify, the fuzzy parameter m_2 (according to the available data, if any) is determined, and the m_1 and m_3 values are determined using MATLAB software. For this reason, for setting these parameters, we only need to mention the value of m_2 . The following values are considered in the generation of sample problems.

In all periods for the demand amount of customer pick-up and delivery, m_2 , based on the data in the EVRPTW benchmark, is the same and equal to the existing demand. Charging capacity, vehicle load capacity, the distance between nodes, and charge consumption rate are considered based on data in the EVRPTW benchmark. Also, among the mentioned parameters, load capacity, charge consumption rate, and charge capacity is fuzzy as described. Travel time between nodes is proportional to their distance and is considered uniform [1....20]. The soft and hard time window is determined based on the data of ready time values and due date values in the EVRPTW benchmark. The ready time is considered equal to the low limit of the soft time window, and the due date is the up limit of the hard time window. Product weight is considered equal to 5. The costs of establishing a depot and a charging station are generated in a uniform range [1000000....10000000], and the fixed and variable costs of using the vehicle are estimated at 2000000 and 100, respectively. Finally, the delay penalty is a fuzzy number from (m_1, m_2, m_3) that m_2 is generated in a uniform range [100....500]. The value of β for ranking the fuzzy numbers is considered 0.8.

5.4. Model validation results

To evaluate the model validity, first, the two-objective model was converted into a single-objective model using the LP-Metric method. Then, the single-objective model was solved in the Gomez software environment for small-size problems. The LP-Metric method was used in the present study. It is one of the most popular methods for multi-objective problems. We always try to minimize the deviations of the objective functions from their optimal value. In this method, the individual responses for optimizing each objective function were first calculated, then the following objective function was minimized.

$$\min z = [w^*(f_1(x) - f_1(x^*)) / f_1(x^*)] + [(1 - w^*)(f_2(x) - f_2(x^*)) / f_2(x^*)] \quad (51)$$

In the above Equation, $f_1(x^*)$ is the optimal value obtained from solving the model considering the first objective function. The $f_2(x)$ is the value of the second objective function based on the optimal solution of solving the model with only the first objective function. The $f_2(x^*)$ is the optimal value obtained from solving model considering the second objective function. The $f_1(x)$ is the value of the first objective function based on the optimal solution of solving the model with only the second objective function. Also, w^* is the weight of the objective function. The proposed model was solved by GAMS coding software and BARON solver. Here, the value of P was considered 1, and the weight of the objectives was considered the same and equal to 0.5. Small-size problems are solved by the MOPSO algorithm and Gomez software to optimize the LP-Metric method. The results of Gomez were evaluated based on the variables values and the constraints. The study results confirmed the feasibility and validity of the model. The results of solving the model by the proposed algorithm and Gomez software are shown in Table 3.

Table 3. Solving results for sample problems of small size

NSGA-II				MOPSO				GAMS			
Time	z	$f_2(x^*)$	$f_1(x^*)$	Time	z	$f_2(x^*)$	$f_1(x^*)$	Time	Z	$f_2(x^*)$	$f_1(x^*)$
0.05	0.183	7243.2	124440.5	0.065	0.367	8031.9	116300.6	0.001	0.476	3380.5	478213.3
0.07	0.145	8397.9	187342.1	0.07	0.269	8263.5	194291.1	0.004	0.236	3876.4	480561.7
0.07	0.284	8149.5	272664.7	0.07	0.284	8149.5	272664.7	0.012	0.456	4069.1	532906.1

According to Table 3 and based on the objective function values, the output solutions for each problem were non-dominated relative to each other. In some problems, the objective function values resulting from the genetic algorithm and the particle swarm algorithm were similar.

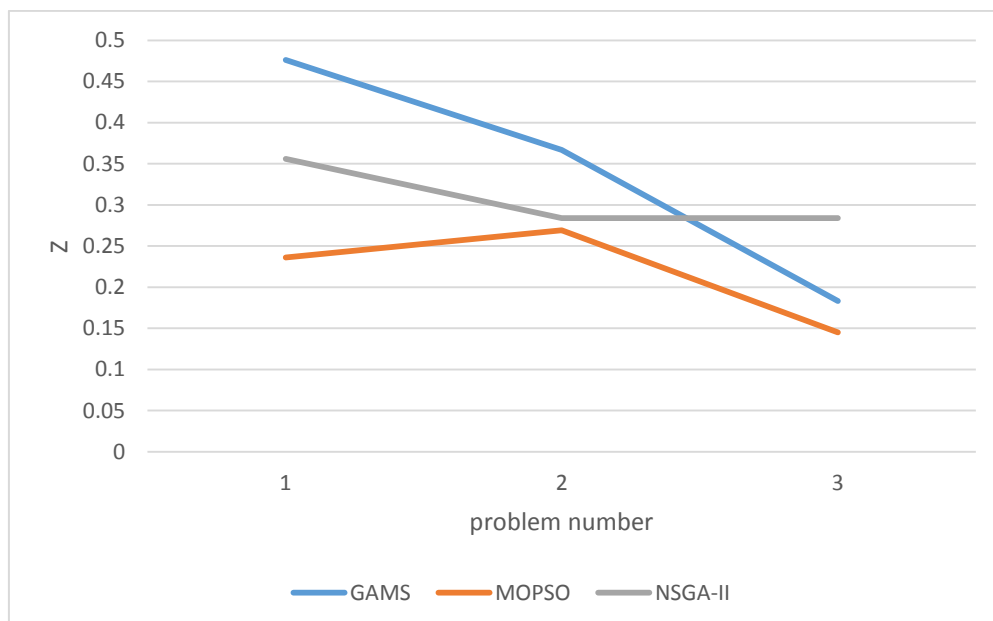


Fig.4. Comparison of objective functions values for problems of small size

On the other hand, comparing the objective function value of the LP-Metric method showed that in some cases, the Gomez software performed better than the two algorithms, whereas in other cases, the performance of the two solution algorithms was better. Considering that the results of Gomez are non-dominated over the particle swarm algorithm and genetics algorithm and the solution of the two algorithms are non-dominated relative to each other, the quality of results obtained in the three methods was at the same level. Hence, it can be concluded that the solution algorithm can solve the model efficiently and converge towards the optimal solution.

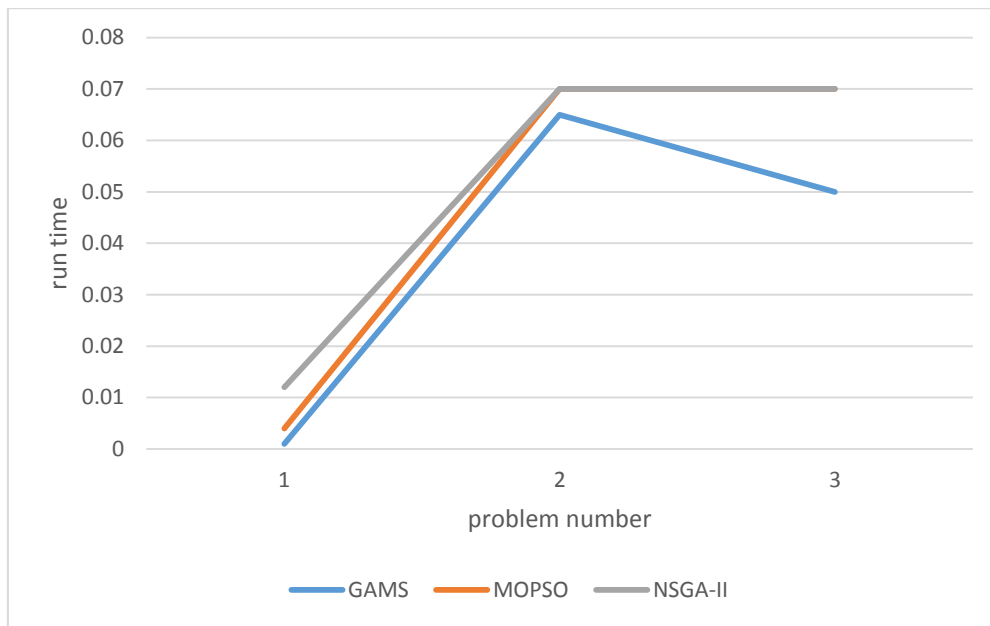


Fig.5. Comparison of performance time for problems of small size

5.5. Performance results

In this section, 30 problems were selected among the experimental problems in the EVRPTW benchmark and solved using the MOPSO and NSGA-II algorithms. Results obtained from informing the two algorithms according to the comparative indices are shown in the following Tables.

Table 4-a. Results of solving EVRPTW sample problems

Prob	MOPSO					NSGA-II				
	Quality metric	Spacing metric	Diversity metric	CPU TIME	Number of Pareto archive responses	Quality metric	Spacing metric	Diversity metric	CPU TIME	Number of Pareto archive responses
C101	94.3	1.12	2686.4	220.7	87	5.7	0.61	1511.2	80.7	60
C102	97.1	0.94	2385.6	230.1	67	2.9	0.70	1741.2	135.4	59
C103	72.9	0.71	2923.6	247.1	77	27.1	0.44	2441.3	119.4	78
C104	100	0.85	2256.7	223.1	85	0	0.60	1437.9	147.9	88
C105	88.6	0.76	2854.1	284.4	93	11.4	0.33	1607.5	141.5	94

C106	100	0.92	2250.2	219.5	95	0	0.57	1349.4	160.4	62
C107	77.6	0.69	2472.2	222.6	79	22.4	0.38	2035.1	159.1	80
C108	85.9	0.86	2780.7	217.1	63	14.1	0.54	2152.1	103.4	76
C109	98.7	0.80	2966.3	222.8	63	1.3	0.57	1708.5	139.9	57
C201	100	0.75	2127.4	243.6	67	0	0.54	1696.5	89.8	70

Table 4-b. Results of solving EVRPTW sample problems

Prob	MOPSO					NSGA-II				
	Quality metric	Spacing metric	Diversity metric	CPU TIME	Number of Pareto archive responses	Quality metric	Spacing metric	Diversity metric	CPU TIME	Number of Pareto archive responses
r101	73.8	0.78	3145.3	231.1	91	26.2	0.58	1652.9	133.7	63
r102	100	1.2	2960.9	292.3	67	0	0.45	2110.8	144.4	89
r103	100	0.79	2614.1	243.1	90	0	0.41	2077.5	120.1	69
r104	84.1	0.68	2553.1	218.5	67	15.9	0.56	1701.1	117.9	78
r105	93.8	0.93	2566.1	290.5	94	6.2	0.61	1035.6	160.5	64
r106	73.4	0.81	2397.6	297.9	71	26.6	0.73	1211.3	130.9	81
r107	82.1	1.1	2640.2	243.9	65	17.9	0.64	1282.9	131.8	68
r108	100	0.90	2642.9	211.1	67	0	0.67	1914.2	155.9	83
r109	93.6	0.88	3011.2	225.8	82	6.4	0.52	1877.7	150.5	85
r110	100	0.94	2983.8	240.9	76	0	0.56	1684.4	127.7	87

Table 4-c. Results of solving EVRPTW sample problems

Prob	MOPSO					NSGA-II				
	Quality metric	Spacing metric	Diversity metric	CPU TIME	Number of Pareto archive responses	Quality metric	Spacing metric	Diversity metric	CPU TIME	Number of Pareto archive responses
rc101	89.3	0.52	2803.2	259.5	71	10.7	0.46	1449.2	88.4	75
rc102	70.1	1.22	2484.3	226.2	90	29.9	0.34	1550.7	93.9	60
rc103	95.3	0.67	3003.9	260.3	80	4.7	0.43	1587.3	158.7	66
rc104	100	1.06	2669.4	271.1	79	0	0.64	1435.1	72.9	94
rc105	90.1	1.1	2450.8	222.2	94	9.9	0.49	1528.3	118.9	63
rc106	92.5	1.1	3156.8	211.7	68	7.5	0.49	1740.4	86.7	90
rc107	92.1	0.56	3081.1	229.7	87	7.9	0.44	1835.9	167.8	79

rc201	81.1	0.76	2690.2	231.9	87	18.9	0.62	1283.6	141.2	97
rc202	89.3	0.64	2776.9	242.4	72	10.7	0.45	1577.5	120.4	60
rc203	74.3	1.3	2734.4	250.8	80	25.6	0.66	1337.6	117.1	75

According to the above Tables, in solving the electric vehicles location-routing problem by considering multi-depots, time windows, and simultaneous pick-up and delivery, the MOPSO algorithm in all states has a higher ability than the NSGA-II algorithm to generate solutions with higher quality. The MOPSO algorithm can generate solutions with higher scatter than the NSGA-II algorithm. In other words, the MOPSO algorithm has more ability for exploring and extracting the solution feasible region than the NSGA-II algorithm. The NSGA-II algorithm generates more uniform solutions than the MOPSO algorithm. In other words, the NSGA-II algorithm searches the solutions region with more uniformity.

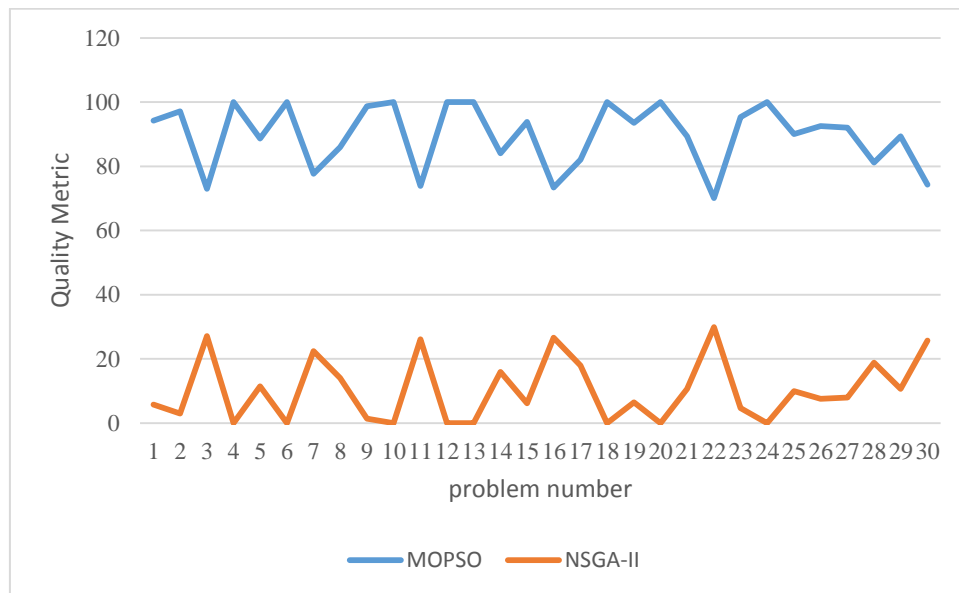


Fig.6. Comparison of quality indice of two algorithms

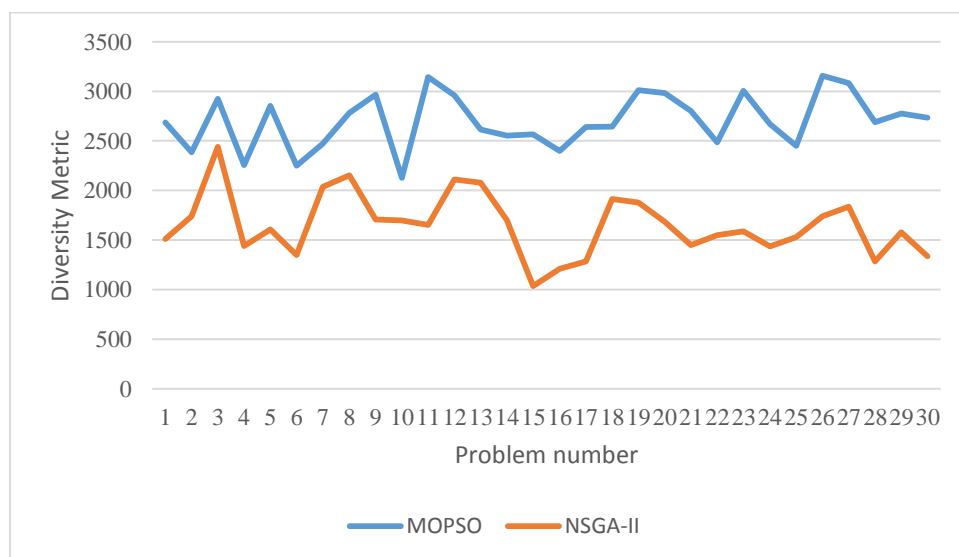


Fig.7. Comparison of scatter indice of two algorithms

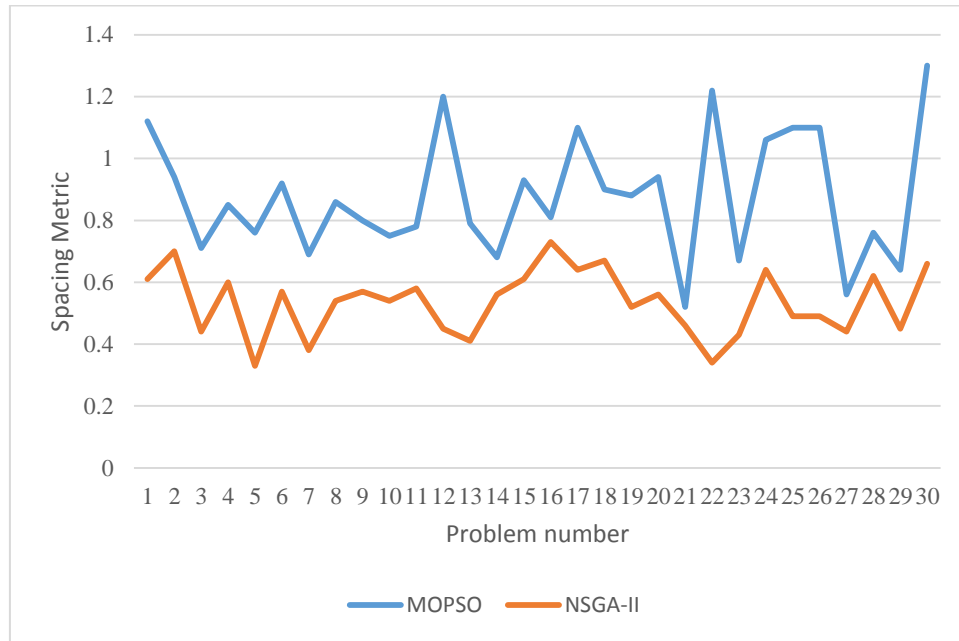


Fig.8. Comparison of uniformity indice of two algorithms

In addition, a comparison of the performing times of the two algorithms to solve the experimental problems is shown in Fig. 9. It shows that the problem-solving time by the MOPSO algorithm is longer than the NSGA-II algorithm in all cases, which means that the MOPSO algorithm needs more time to solve these problems.

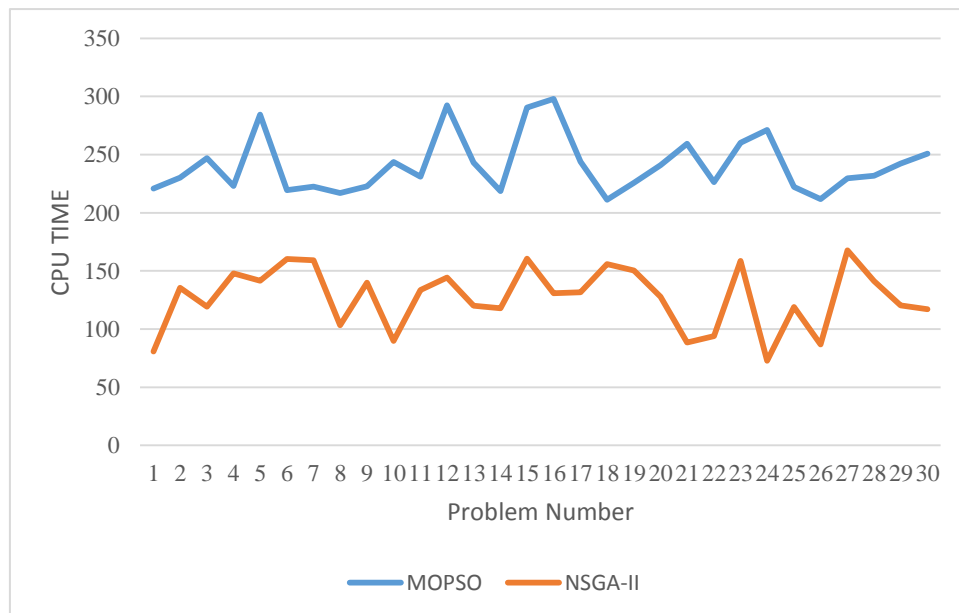


Fig.9. Comparison of performing time (sec)

5.6. Statistical analysis of comparing two algorithms

To solve the proposed model in this study, a multi-objective particle swarm algorithm and NSGA-II algorithm were used. Comparing the results of solving the sample problems by two algorithms was performed based on comparative quality indices, scatter, and uniformity. In this section, by assigning the appropriate assumptions, we studied the differences between the results of the two algorithms based on statistical analysis.

Assumption 1: There is a considerable difference between the quality indices of the MOPSO algorithm and the NSGA-II algorithm.

Table 5. Test results of the first assumption

Average error	Standard deviation	Average	Sample volume	Sample
1.8	9.8	89.7	30	MOPSO
1.8	9.8	10.3	30	NSGA-II

Average error	Standard deviation	Average	Sample volume	Sample
1.8	9.8	89.7	30	MOPSO
1.8	9.8	10.3	30	NSGA-II

As shown in Table 5, the average of the two groups is very different from each other. Also, the statistic value is outside the confidence range. Thus, the assumption H0 is rejected, and the assumption H1 is accepted. This means a considerable difference between the quality indice of the MOPSO and the NSGA-II algorithms.

Assumption 2: There is a considerable difference between the quality indice of MOPSO and NSGA-II algorithms.

Table 6. Test results of the second assumption

Average error	Standard deviation	Average	Sample volume	Sample
0.036	0.199	0.87	30	MOPSO
0.019	0.105	0.53	30	NSGA-II

	t	Freedom degree	Level of significance	Mean difference	95% confidence range	
					Lower	Upper
MOPSO	22.712	29	.000	.82800	.7534	.9026
NSGA-II	25.147	29	.000	.48433	.4449	.5237

As shown in Table 6, the average of the two groups is very different from each other. Also, the statistic value is outside the confidence range. Thus, the assumption H0 is rejected, and the assumption H1 is accepted. This means a considerable difference between the uniformity indice of the MOPSO and the NSGA-II algorithms.

Assumption 3: There is a considerable difference between the scatter indice of MOPSO and NSGA-II algorithms.

Table 7. Test results of the third assumption

Average error	Standard deviation	Average	Sample volume	Sample
50.2	274.8	2702.3	30	MOPSO
56.9	311.7	1651.8	30	NSGA-II

Average error	Standard deviation	Average	Sample volume	Sample
50.2	274.8	2702.3	30	MOPSO
56.9	311.7	1651.8	30	NSGA-II

As shown in Table 7, the average of the two groups is very different from each other. Also, the statistic value is outside the confidence range. Thus, the assumption H_0 is rejected, and the assumption H_1 is accepted. This means a considerable difference between the scatter indice of the MOPSO and the NSGA-II algorithms.

Assumption 4: There is a considerable difference between the performing time of MOPSO and NSGA-II algorithms.

Table 8. Test results of the fourth assumption

Average error	Standard deviation	Average	Sample volume	Sample
4.5	24.8	241.1	30	MOPSO
4.9	26.5	127.2	30	NSGA-II

t		Freedom degree	Level of significance	Mean difference	95% confidence range	
					Lower	Upper
MOPSO	53.128	29	.000	241.01333	231.7353	250.2914
NSGA-II	26.256	29	.000	127.18000	117.2734	137.0866

As shown in Table 8, the average of the two groups is very different from each other, and the statistics amount is outside of confidence distance. Thus, the assumption H_0 is rejected, and the assumption H_1 is accepted. This means that there is a considerable difference between the performing time of the MOPSO and the NSGA-II algorithms.

6. Conclusions

The present study proposed a mathematical model for the location-routing of electric vehicles by considering hard and soft time windows and multi-depots. Here, the capacity of the vehicle was limited, and the charge consumption amount was dependent on the load of the vehicle. Vehicles were fully charged at charging stations. In this paper, a

fuzzy-mixed-integer programming model was first presented, and the Jimenez method was used to convert the fuzzy model to a definite model. The mathematical model had two objectives, (i) minimizing costs and (ii) minimizing penalties for late delivery of commodities or services to customers. After model designing, a combined multi-objective particle swarm algorithm and NSGA-II algorithm were proposed for solving the model. In order to evaluate the validity of the model and algorithm, the presented model was performed and solved in the Gomez precision software for small size problems, and the obtained results were compared to the solution algorithms results. Then, the problem-solving results by the proposed algorithm were compared to the solution results by the NSGA-II algorithm based on comparative indices of quality, scatter, and uniformity.

In general, the results of solving the model showed that:

- The model evaluation and solution algorithms confirmed the model validity and had a feasible area. Also, the solution algorithms can solve the model effectively and converge towards the optimal solution.
- Comparative results showed that the MOPSO algorithm in all cases has a higher ability than the NSGA-II algorithm to generate solutions with higher quality and more scatter. Also, the NSGA-II algorithm has more uniform solutions than the MOPSO algorithm. In addition, the results showed that the performing time of the multi-objective MOPSO algorithm is longer than the NSGA-II algorithm.
- There is a considerable difference between the quality indice of MOPSO and NSGA-II algorithms.
- There is a considerable difference between the uniformity indice of MOPSO and NSGA-II algorithms.
- There is a considerable difference between the scatter indice of MOPSO and NSGA-II algorithms.
- There is a considerable difference between the performing time of MOPSO and NSGA-II algorithms.

7. Recommendations for future research

- Designing the model in the full-charge mode and half-charge mode.
- Considering the vehicles with combined combustion and electric engines.
- Considering probabilistic and fuzzy parameters together to apply uncertainty.
- Using the other algorithms such as Whale Optimization Algorithm (WOA) or Imperialist Competitive Algorithm (ICA) to solve the model.

Declarations

Source of Funding

This research did not receive any grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing Interests Statement

The authors declare no competing financial, professional and personal interests.

Consent for publication

Authors declare that they consented for the publication of this research work.

References

- [1] Juan, A.A., J. Goentzel, and T. Bektaş. (2014). Routing fleets with multiple driving ranges: Is it possible to use greener fleet configurations?, *Applied Soft Computing*, 21:84-94.
- [2] Touati-Moungla, N., & Jost, V. (2012). Combinatorial optimization for electric vehicles management. *Journal of Energy and Power Engineering*, 6(5).
- [3] Zhang, Shuzhu, et al. (2014). Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem. *Transportation Research Part D: Transport and Environment*, 31, pp. 8599.
- [4] Elhedhli, Samir, and Ryan Merrick. (2012). Green supply chain network design to reduce carbon emissions. *Transportation Research Part D: Transport and Environment*, 17.5, pp. 370-379.
- [5] Erdoğan, Sevgi, and Elise Miller-Hooks. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48.1, pp. 100-114.
- [6] Schneider, Michael, Andreas Stenger, and Dominik Goeke. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48.4, pp. 500-520, 2014.
- [7] Felipe, Ángel, et al. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*. 71, pp. 111-128.
- [8] Goeke, Dominik, and Michael Schneider. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245.1, pp. 81-99.
- [9] Koç, Çağrı, and Ismail Karaoglan. (2016). The green vehicle routing problem: A heuristic-based exact solution approach. *Applied Soft Computing*, 39, pp. 154-164.
- [10] Keskin, Merve, and Bülent Çatay. (2016). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65, pp. 111-127.
- [11] Roberti, Roberto, and Min Wen. (2016). The Electric Traveling Salesman Problem with Time Windows. *Transportation Research Part E: Logistics and Transportation Review*, 89, pp. 32-52.
- [12] Hiermann, Gerhard, et al. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252.3, pp 995-1018.
- [13] Doppstadt, C., A. Koberstein, and D. Vigo. (2016). The Hybrid Electric Vehicle–Traveling Salesman Problem. *European Journal of Operational Research* 253.3, pp 825-842.
- [14] Schiffer, M., Walther, G. (2017). The electric location routing problem with time windows and partial recharging. *European Journal of Operational Research*, 260, pp. 995-1013.
- [15] Camilo Paza, J., Granada-Echeverria, M., Willmer Escobar, J. (2018). The multi-depot electric vehicle location routing problem with time windows. *International Journal of Industrial Engineering Computations*, 9, pp. 123–136.

- [16] Jiménez, M., et al., (2007). Linear programming with fuzzy parameters: an interactive method resolution. *European Journal of Operational Research*, 177(3): pp. 1599-1609.
- [17] Deb, K., et al. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2): pp. 182-197.
- [18] Tavakkoli-Moghaddam, R., M. Azarkish, and A. Sadeghnejad-Barkousaraie, (2011). A new hybrid multi-objective Pareto archive PSO algorithm for a bi-objective job shop scheduling problem. *Expert Systems with Applications*, 38(9): pp. 10812-10821.
- [19] Rohmer, S. U. K., van Lemmen-Gerdessen, J. C., & Claassen, G. D. H. (2019). A Location-Routing Problem with Delivery Options and Time-Windows for the Last-Mile Delivery of Fresh Products. Abstract from Workshop of the EURO Working Group on Vehicle Routing and Logistics optimization (VeRoLog), Sevilla, Spain.
- [20] Almouhanna, Abdullah & Quintero-Araujo, Carlos & Panadero, Javier & Juan, Angel & Khosravi, Banafsheh & Ouelhadj, Djamil. (2019). The Location Routing Problem using Electric Vehicles with constrained Distance. *Computers & Operations Research*. 115. 10.1016/j.cor.2019.104864.
- [21] Zheng, Zilai & Morimoto, Takehiro & Murayama, Yuji. (2020). Optimal Location Analysis of Delivery Parcel-Pickup Points Using AHP and Network Huff Model: A Case Study of Shiweitang Sub-District in Guangzhou City, China. *ISPRS International Journal of Geo-Information*, 9. 193.
- [22] Ghobadi, A., Tavakkoli Moghadam, R., Fallah, M., Kazemipoor, H. (2021). Multi-depot electric vehicle routing problem with fuzzy time windows and pickup/delivery constraints. *Journal of Applied Research on Industrial Engineering*, 8(1), pp. 1-18. Doi: 10.22105/jarie.2021.231764.1165.